

A SIMULINK LIBRARY FOR INVERTED PENDULA MODELING AND SIMULATION

Slávka Jadlovská, Anna Jadlovská

Department of Cybernetics and Artificial Intelligence, Faculty of Electrical Engineering and Informatics, Technical University of Košice, Slovakia

Abstract

This paper aims to provide an insight into the functionality of the *Inverted Pendula Modeling and Control*, a custom, thematic *Simulink* block library which was designed to provide software support for the analysis and synthesis of inverted pendula systems. In addition, crucial modeling procedures such as the derivation of motion equations for inverted pendula systems and symbolic linearization with respect to a given equilibrium point are presented in form of symbolic algorithms, generalized for a system of n inverted pendula.

1 Introduction

Inverted pendula systems are a significant group of mechanical systems used in control education with a number of practical applications, such as those listed in [1][2]:

- simulation of the unstable system of a human or robotic upper limb if the centre of pressure is placed below its centre of gravity
- modeling a human or a robot standing upright
- simulation of a space shuttle or a rocket taking off
- missile guidance if thrust is actuated at the bottom of a tall vehicle

Within the MATLAB/*Simulink* environment, block libraries (blocksets) represent the modern object-oriented, event-flow-driven, user-friendly problem-solving approach. The *Simulink Library Browser* allows a number of built-in (Toolbox) libraries to be accessed in order to solve and simulate various scientific and technical issues by means of block interconnecting. Furthermore, to provide a wider variety of problems with such user flexibility, custom masked blocks may be created and grouped into user-designed libraries.

In order to cope with the modeling- and control-related issues concerning the systems of n inverted pendula on a cart, a block library entitled *Inverted Pendula Modeling and Control* (*IPMaC* for short) was designed that can be fully integrated into the *Simulink Library Browser* and used identically to any built-in *Simulink* blockset. An important part of the library is the *Demo Simulations* section, which represents a collection of links to simulation schemes whose purpose is to solve various analysis- and synthesis-related problems.

The enclosed MATLAB functions (*m-files*) make use of the *Symbolic Math Toolbox* to generalize the process of modeling for a system of n inverted pendula. Compared to the often used, but error-prone Newton force summation, the easily algorithmizable approach based on Lagrange mechanics points out the analogy found while creating the mathematical models of these systems for a changing n . This approach was thus employed to develop a general MATLAB algorithm that derives the symbolic equations of motion for any given n . Following the similar philosophy, the procedure of linear approximation of a nonlinear system in a given equilibrium point was also made into a MATLAB algorithm.

2 The Structure of the *IPMaC* Block Library

The installation of the library itself into the *Simulink Library Browser* consists of unzipping the *InvPend.zip* package and calling the included installation script *slblocks.m*. Refreshing the tree view in the left panel is then required to reach the situation as in Fig. 1, which shows the final stage of the installation process – the whole sublibrary structure of the *IPMaC* is seen as an integral part of the Browser. This implies that the library is used identically to any of *Simulink* block libraries.

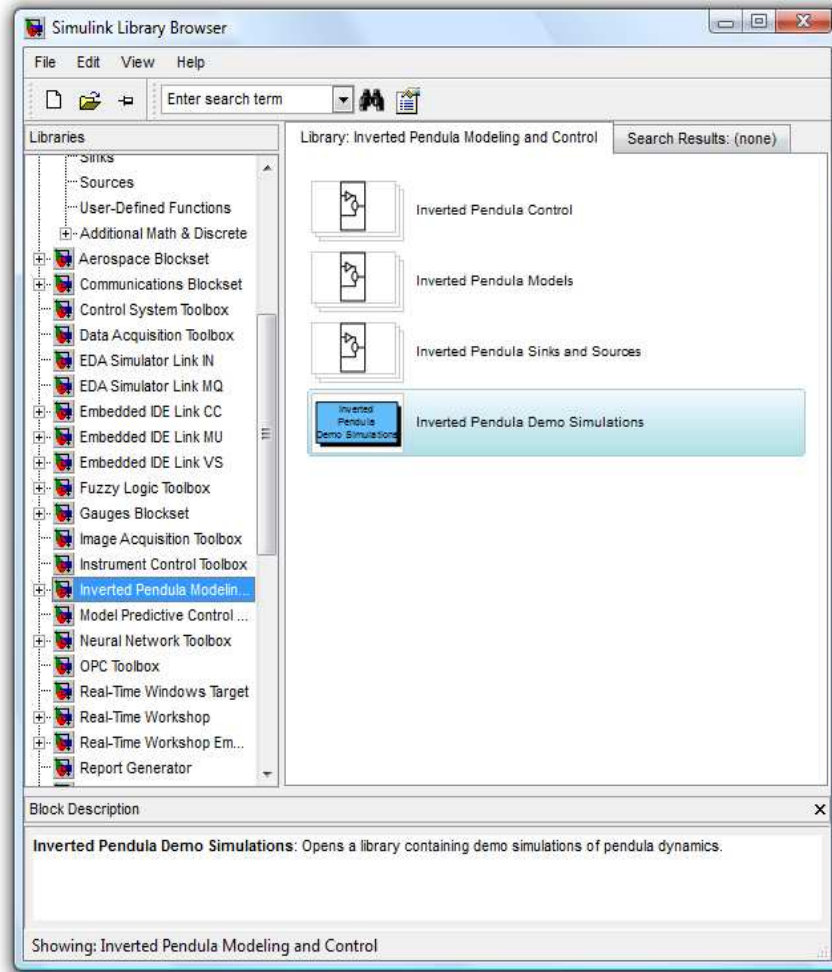


Fig. 1 The IPMaC, installed and active in *Simulink Library Browser*

The sublibraries that the *IPMaC* block library is structured into are:

- *Inverted Pendula Models* sublibrary – contains dynamic-masked simulation models of inverted pendula systems, pre-prepared for use in open-loop analysis as well as state-space control
- *Inverted Pendula Control* sublibrary – contains function blocks that enable the state-space feedback control design
- *Inverted Pendula Sinks and Sources* sublibrary – contains useful custom-designed input/output blocks e.g. the *Impulse* block, which represents the most typical input signal in case of inverted pendula simulation, i.e. a signal constrained in terms of time and amplitude; or the *Scope rad2deg* block, which displays its input signal with an angle meaning in degrees rather than in radians.

The *Demo Simulations* section has the form of a tree structure of links to example simulation schemes that demonstrate the functionality of the simulation blocks. Each demo scheme can be accessed by doubleclicking a block that briefly describes the issue it solves. These range from analyses of the uncontrolled systems' dynamics to various verifications of control algorithms.

3 Mathematical Modeling

3.1 Theoretical Background behind a General Procedure of Finding the Motion Equations

Let us consider a system of n homogenous, isotropic rods which are joint-bound together and attached to an inherently stable moving base. Hence we obtain a nonlinear mechanical SIMO system of

n inverted pendula on a cart. The system's only input is the force acting upon the cart, the $n+1$ outputs are the cart position $[m]$ and pendula angles $[rad]$. If we represent the system's outputs in form of a vector of generalized coordinates

$$\boldsymbol{\theta}(t) = (\theta_0(t) \quad \theta_1(t) \quad \dots \quad \theta_n(t))^T \quad (1)$$

then the system can be mathematically described by the Euler-Lagrange-Rayleigh equations, whose condensed vector form is

$$\frac{d}{dt} \left(\frac{\partial L(t)}{\partial \dot{\boldsymbol{\theta}}(t)} \right) - \frac{\partial L(t)}{\partial \boldsymbol{\theta}(t)} + \frac{\partial D(t)}{\partial \dot{\boldsymbol{\theta}}(t)} = \boldsymbol{Q}^*(t) \quad (2)$$

where $L(t)$ (Lagrange function) is defined as the difference between the system's kinetic and potential energy, $D(t)$ (Rayleigh, dissipation function) expresses the viscous (friction) forces and $\boldsymbol{Q}^*(t)$ is the vector of *generalized external forces* acting upon the system. The use of Lagrange mechanics therefore transforms the process of deriving the motion equations of the inverted pendula system into the determination of kinetic, potential and dissipation energies related to the cart and all pendula. Using well-known formulas, general relations that describe the energetic balances of the cart and i -th pendulum in an inverted pendula system were derived and can be found in [1].

These computations are at the base of the general MATLAB algorithm (*invpendderiv.m*) that uses the *Symbolic Math Toolbox* to return the mathematical model (i.e. n differential equations of motion) of the system; the number of pendula (n) is specified as the function parameter. The command window output of the `invpendderiv(1)` function call is listed below. In addition to the eventual symbolic motion equations in the "pretty" form, all physically significant steps of the derivation process can be tracked. The equations are generated in the simplified and rearranged form, which is equivalent to the most likely form obtained by manual derivation.

```

eq=invpenderv(1)
Cart Coordinate:
x0 =
fi0
Pendulum 1 Coordinates:
x1 =
fi0+1/2*l1*sin(fi1)
y1 =
1/2*l1*cos(fi1)
Cart Velocity:
v0 =
dfi0
Pendulum 1 Velocity:
v1 =
1/2*(4*(dfi0+1/2*l1*dfil*cos(fi1))^2+l1^2*dfil^2*sin(fi1)^2)^(1/2)
Cart Potential Energy:
Ep0 =
0
Cart Kinetic Energy
Ek0 =
1/2*m0*dfi0^2
Pendulum 1 Potential Energy:
Epl =
1/2*m1*g*l1*cos(fi1)
Pendulum 1 Kinetic Energy:
Ekl =
1/2*m1*((dfi0+1/2*l1*dfil*cos(fi1))^2+1/4*l1^2*dfil^2*sin(fi1)^2)+1/2*JT1*dfil^2
Dissipative Energy:0
Ed0 =
1/2*delta0*dfi0^2
Dissipative Energy:1

```

```

Ed1 =
1/2*delta1*dfl1^2
Lagrange Function:
1/2 m1 dfl0 l1 dfl1 cos(fl1) - 1/2 m1 g l1 cos(fl1) + 1/2 m0 dfl0
2 2 2 2
+ 1/2 m1 dfl0 + 1/2 JT1 dfl1 + 1/8 m1 l1 dfl1
Dissipative Function:
1/2 delta0 dfl0 + 1/2 delta1 dfl1
EQUATIONS OF MOTION:
(m0 + m1) |--- fl0(t)| + 1/2 m1 l1 |--- fl1(t)| cos(fl1(t))
| 2 | | 2 |
\dt / \dt /
- 1/2 m1 l1 |-- fl1(t)| sin(fl1(t)) + delta0 |-- fl0(t)|
\dt / \dt /
(JT1 + 1/4 m1 l1 ) |--- fl1(t)| + 1/2 m1 |--- fl0(t)| l1 cos(fl1(t))
| 2 | | 2 |
\dt / \dt /
- 1/2 m1 g l1 sin(fl1(t)) + delta1 |-- fl1(t)|
\dt /
eq =
[ (m0+m1)*d2fl0+1/2*m1*l1*d2fl1*cos(fl1)-1/2*m1*l1*dfl1^2*sin(fl1)+delta0*dfl0,
(JT1+1/4*m1*l1^2)*d2fl1+1/2*m1*d2fl0*l1*cos(fl1)-1/2*m1*g*l1*sin(fl1)+delta1*dfl1]

```

To verify the credibility of the algorithm, the automatically received nonlinear differential equations were, in both cases, compared to equations derived manually (in [3],[4] for $n=1$, in [5],[6] for $n=2$), obtaining identical results.

3.2 Simulation Models of Single and Double Inverted Pendulum

The sublibrary of the *IPMaC* contains simulation models of the single ($n=1$) and double ($n=2$) inverted pendulum in form of atomic library blocks: *Single Inverted Pendulum on a Cart* and *Double Inverted Pendulum on a Cart* (*SIPoac* and *DIPoac* for short); both with their own icon and parametric mask. The designed blocks can be used to compensate for an unavailable real model.

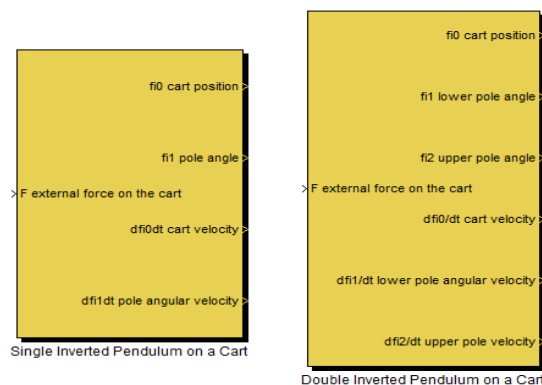


Fig. 2 Simulink blocks of inverted pendula models included in the *IPMaC* library

The block mask of each implemented system (*Edit – Mask Subsystem*) encompasses the features such as changing the system’s parameters, specifying the initial conditions (thus enabling the initial

deflection analysis), enabling or disabling the input force and adjusting the number of outputs, which is equivalent to equipping a real model with sensors. Setting a number and type of the block's inputs and outputs is actually done by scripts that are assigned to the specified parameters within the mask. The purpose of the scripts is to add the *Input/Output Port* block into the scheme whenever the input or one of the outputs is marked as required. Otherwise, they are replaced by *Ground/Terminator* blocks, which basically disable the input/output on a block.

The inner structure of both blocks is composed of logically designed subsystem blocks interconnected with one another with respect to their mutual physical relations. Since each subsystem block corresponds to a nonlinear equation that is part of the system's mathematical model, the mathematical model of the system represents a necessary prerequisite for simulation. It can be concluded from above that the mathematical model of a system of n inverted pendula consists of $n+1$ second-order differential equations that describe the cart and each pendulum.

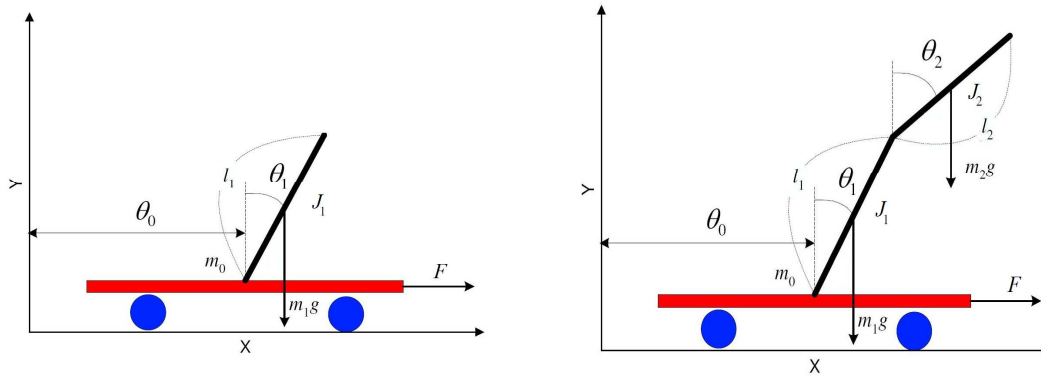


Fig. 3 Single and double inverted pendulum on a cart – scheme and basic nomenclature

By setting $n = 1$, we obtain the *single inverted pendulum* system (Fig. 3, left), which encompasses the cart subsystem:

$$(m_0 + m_1)\ddot{\theta}_0(t) + \delta_0\dot{\theta}_0(t) + \frac{1}{2}m_1l_1(\ddot{\theta}_1(t)\cos\theta_1(t) - \dot{\theta}_1^2(t)\sin\theta_1(t)) = F(t) \quad (3)$$

and the pendulum subsystem:

$$J_1\ddot{\theta}_1(t) + \delta_1\dot{\theta}_1(t) + \frac{1}{2}m_1l_1\ddot{\theta}_0\cos\theta_1(t) - \frac{1}{2}m_1gl_1\sin\theta_1(t) = 0 \quad (4)$$

where m_0 is the cart mass, m_1 is the pendulum mass, l_1 is the pendulum length, δ_0 is the friction coefficient of the cart against the surface, δ_1 is the damping constant of the pendulum and $J_1 = \frac{1}{3}m_1l_1^2$ is the pendulum's moment of inertia with respect to the pivot.

Analogously, calling the *invpenderv.m* with $n = 2$ yields the mathematical model of the *double inverted pendulum* (Fig. 3, right), which is composed of the cart subsystem:

$$(m_0 + m_1 + m_2)\ddot{\theta}_0(t) + \delta_0\dot{\theta}_0(t) + \left(\frac{1}{2}m_1l_1 + m_2l_1\right)(\ddot{\theta}_1(t)\cos\theta_1(t) - \dot{\theta}_1^2(t)\sin\theta_1(t)) + \frac{1}{2}m_2l_2(\ddot{\theta}_2(t)\cos\theta_2(t) - \dot{\theta}_2^2(t)\sin\theta_2(t)) = F(t) \quad (5)$$

the lower pendulum subsystem:

$$(J_1 + m_2l_1^2)\ddot{\theta}_1(t) + (\delta_1 + \delta_2)\dot{\theta}_1(t) - \delta_2\dot{\theta}_2(t) + \left(\frac{1}{2}m_1l_1 + m_2l_1\right)\ddot{\theta}_0(t)\cos\theta_1(t) + \frac{1}{2}m_2l_1l_2(\ddot{\theta}_2(t)\cos(\theta_1(t) - \theta_2(t)) + \dot{\theta}_2^2(t)\sin(\theta_1(t) - \theta_2(t))) - \left(\frac{1}{2}m_1 + m_2\right)gl_1\sin\theta_1(t) = 0 \quad (6)$$

and the upper pendulum subsystem:

$$\begin{aligned}
& J_2 \ddot{\theta}_2(t) + \delta_2 (\dot{\theta}_2(t) - \dot{\theta}_1(t)) + \frac{1}{2} m_2 l_2 \ddot{\theta}_0(t) \cos \theta_2(t) + \\
& + \frac{1}{2} m_2 l_1 l_2 (\ddot{\theta}_1(t) \cos(\theta_1(t) - \theta_2(t)) - \dot{\theta}_1^2(t) \sin(\theta_1(t) - \theta_2(t))) - \frac{1}{2} m_2 g l_2 \sin \theta_2(t) = 0
\end{aligned} \tag{7}$$

where m_0 is the cart mass, m_1, m_2 are the pendula masses, l_1, l_2 are the pendula lengths, δ_0 is the friction coefficient of the cart, δ_1, δ_2 are the damping constants of the pendula and $J_1 = \frac{1}{3} m_1 l_1^2$, $J_2 = \frac{1}{3} m_2 l_2^2$ are the moment of inertia of the pendula with respect to the pivot points.

As an illustration, Fig. 4 and Fig. 5 depict the inner structure of subsystem blocks *Cart* and *Pendulum* within the *Single Inverted Pendulum on a Cart* block, both of which represent an exact transformation of nonlinear equations into *Simulink* block diagrams.

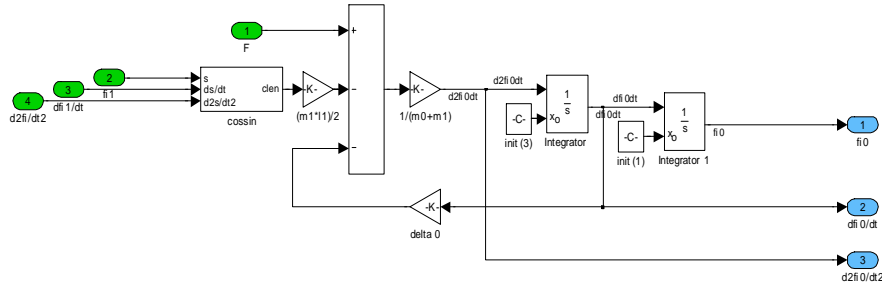


Fig. 4 The *Cart* subsystem within the function block *Single Inverted Pendulum on a Cart*

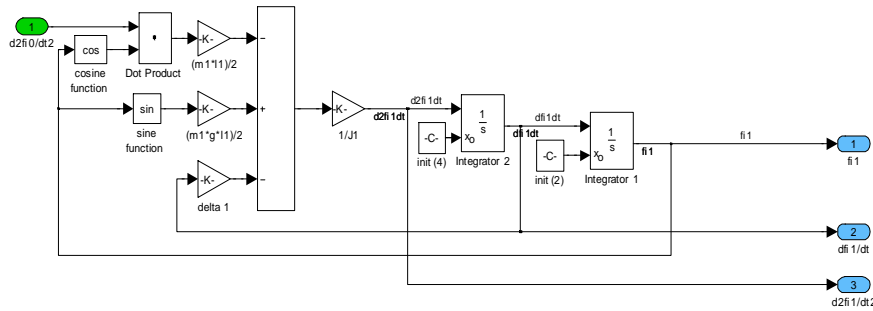


Fig. 5 The *Pendulum* subsystem within the function block *Single Inverted Pendulum on a Cart*

Thanks to the *invpendderiv.m*, the derivation of the motion equations for any number of pendula has been made automatic and further expansion of the *Inverted Pendula Models* section (e.g. by creating a triple or quadruple inverted pendulum model) is straightforward.

4 State-Space Control Algorithms

It is well-known from experience that independently of the initial state, an inverted pendula system reaches the stable equilibrium point in which all pendula are pointing downward. The control objective will therefore be to stabilize the pendulum (pendula) in the upright, unstable position. Problems such as nonzero initial conditions, compensation of disturbance signal, tracking a reference position of the cart or a combination of the three were dealt with.

4.1 Linear Approximation of Inverted Pendula Systems

Since the use of linear methods of synthesis involves the linear approximation of the original nonlinear dynamic system of inverted pendula, the general procedure of linearization was algorithmized so as to yield the symbolic or numeric form of linear state-space matrices A, b, C related to a user-chosen equilibrium point of a given nonlinear, physically realizable system.

As the order of any system of n inverted pendula on a cart is $2n+2$, a state vector in the following form was introduced to describe the system:

$$\mathbf{x}(t) = (\boldsymbol{\theta}(t) \quad \dot{\boldsymbol{\theta}}(t))^T = (x_1(t) \quad x_2(t) \quad \dots \quad x_{2n+2}(t))^T \tag{8}$$

and the force acting upon the cart was logically defined as the only input of the system: $u(t) = F(t)$.

Rewriting the Lagrange motion equations into the minimal *ODE* form

$$\mathbf{M}(\boldsymbol{\theta}(t))\ddot{\boldsymbol{\theta}}(t) + \mathbf{N}(\boldsymbol{\theta}(t), \dot{\boldsymbol{\theta}}(t))\dot{\boldsymbol{\theta}}(t) + \mathbf{P}(\boldsymbol{\theta}(t)) = \mathbf{V}(t) \quad (9)$$

makes it possible to isolate the derivative $\ddot{\boldsymbol{\theta}}(t)$ of state-space vector and subsequently to express the system in the nonlinear state-space form:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t), u(t), t) \\ \mathbf{y}(t) &= \mathbf{g}(\mathbf{x}(t), u(t), t) \end{aligned} \quad (10)$$

With use of the Taylor series, we can now create a linear approximation to the whole state equation by substituting $f_i(\mathbf{x}(t), u(t))$ by $f_i^*(\mathbf{x}(t), u(t))$:

$$f_i^*(\mathbf{x}(t), u(t)) \approx f_i(\mathbf{x}_s, u_s) + \sum_{k=1}^{2n+2} \left. \frac{\partial f_i(\mathbf{x}(t), u(t))}{\partial x_k(t)} \right|_{\mathbf{x}_s, u_s} (x_k(t) - x_{ks}) + \left. \frac{\partial f_i(\mathbf{x}(t), u(t))}{\partial u(t)} \right|_{\mathbf{x}_s, u_s} (u(t) - u_s) \quad (11)$$

In the case of the upright position, $\mathbf{x}(t) = \mathbf{x}_s = \mathbf{0}^T$ and $u(t) = u_s = 0$ holds and the state-space description of the physically realizable linearized system is given as

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{b}u(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) \end{aligned} \quad (12)$$

The above-mentioned algorithm of transforming the Lagrange mathematical model into a state-space matrix form is based on the just described process. For instance, the command window output produced by the *matrices_single.m* function which returns the linear state-space matrices of the single inverted pendulum system in the upright position is as follows:

```
[A,b,C,d]=matrices_single
State space matrices
A =
    [ 0 , 0 , 1 , 0 ]
    [ 0 , 0 , 0 , 1 ]
    [      g m1      delta0      delta1      ]
    [ 0 , -3 ----- , -4 ----- , 6 ----- ]
    [      m1 + 4 m0      m1 + 4 m0      11 (m1 + 4 m0) ]
    [      g (m0 + m1)      delta0      (m0 + m1) delta1 ]
    [ 0 , 6 ----- , 6 ----- , -12 ----- ]
    [      11 (m1 + 4 m0)      11 (m1 + 4 m0)      2      ]
    [                                     m1 11 (m1 + 4 m0) ]

b =
    [      0      ]
    [      ]
    [      0      ]
    [      ]
    [      4      ]
    [ ----- ]
    [ m1 + 4 m0 ]
    [      ]
    [      6      ]
    [ - ----- ]
    [ 11 (m1 + 4 m0) ]

C =
    1      0      0      0
    0      1      0      0

d =
    0
    0
```

As in the case of mathematical model derivation, the manually computed results (e.g. from [4]) confirm that the function output is accurate.

4.2 Inverted Pendula Control

At the moment, the *Inverted Pendula Control* sublibrary of the IPMaC provides software support for continuous feedback methods of controller design, once more in form of dynamic-masked custom blocks. Most importantly, it is the *State Space Controller block (SSC)*, which evaluates the relation

$$u(t) = u_f(t) + u_v(t) + d_u(t) = -\mathbf{k}\mathbf{x}(t) + \mathbf{k}_v w(t) + d_u(t) \quad (13)$$

where $u_f(t)$, $u_v(t)$ and $d_u(t)$ stand for the feedback, setpoint and unmeasured disturbance components respectively; \mathbf{k} is the feedback gain which brings the system's state vector to the origin of the state space ([1][6][7]), and \mathbf{k}_v represents the setpoint gain which needs to be applied if a nonzero required value is specified ([7]). The block's dynamic mask (Fig. 6) allows the user to pick the method to determine the feedback gain \mathbf{k} : the pole-placement algorithm or the linear quadratic regulation (LQR) optimal control method are available, both of which are supported by *Control Toolbox* in form of built-in functions (*acker/place*, *lqr*). The corresponding computations are programmed to occur during the initialization of the block, which eliminates the need for an additional m-file. The nonzero setpoint input $w(t)$ and disturbance input $d_u(t)$ may optionally be enabled or disabled so as to adjust the block's appearance to match the control objective.

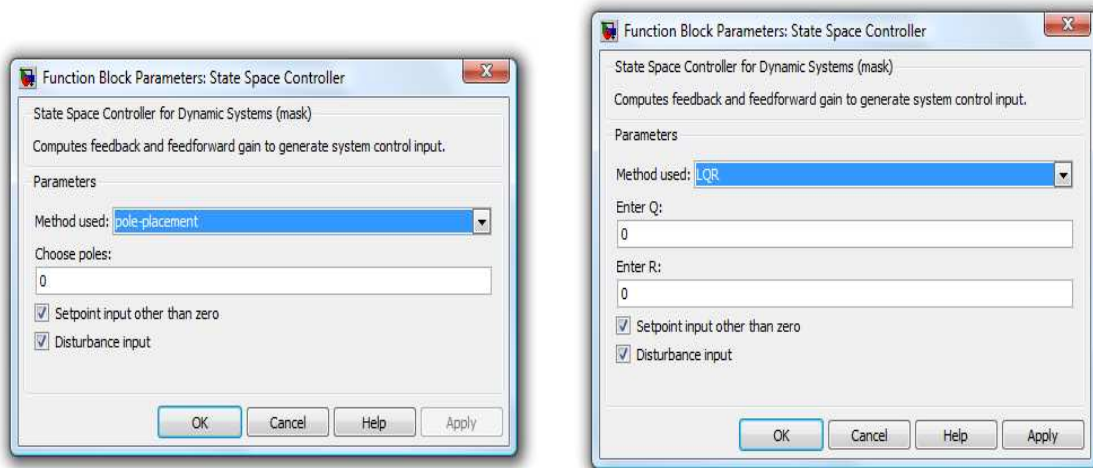


Fig. 6 The dynamically changing State Space Controller dialog box according to the chosen method

The *state estimator block (SE)* represents a model of the estimated system in the form:

$$\dot{\hat{\mathbf{x}}}(t) = \mathbf{A}\hat{\mathbf{x}}(t) + \mathbf{b}u(t) + \mathbf{L}(\mathbf{y}(t) - \mathbf{C}\hat{\mathbf{x}}(t)) \quad (14)$$

where \mathbf{L} is the estimator gain matrix and $\hat{\mathbf{x}}(t)$ is the estimated state vector provided by the *SE*. The estimation process is based on the gradual minimisation of the estimation error $\tilde{\mathbf{x}}(t) = \mathbf{x}(t) - \hat{\mathbf{x}}(t)$ and on the requirement that the time behaviour of the error $\dot{\tilde{\mathbf{x}}}(t) = (\mathbf{A} - \mathbf{L}\mathbf{C})\tilde{\mathbf{x}}(t)$ should stay independent of system parameters.

5 Demo Simulations – Overview

The *Inverted Pendula Demo Simulations* section of the IPMaC is actually a special type of library block. If dragged into a simulation scheme and doubleclicked, it displays a structured set of links to simulation schemes (Fig. 7) which demonstrate the functionality of the blocks described in the previous sections. Making a classic masked block to act as a link to a *.mdl* (or any other) file involves creating an *OpenFcn* callback function within the *Block Properties*.

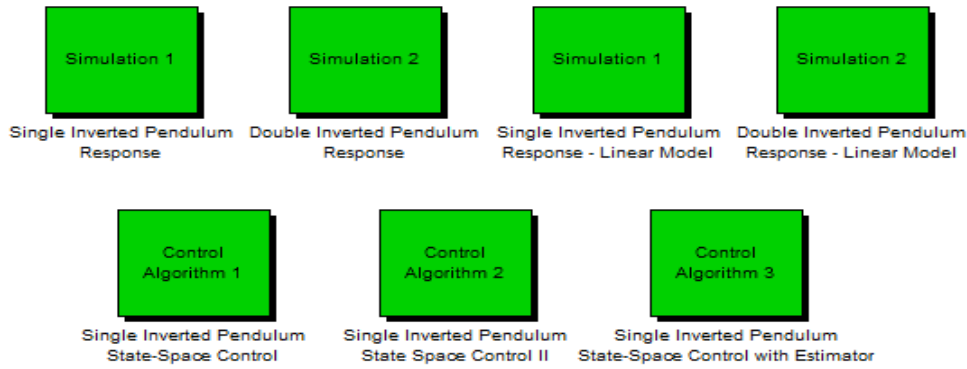


Fig. 7 Structure of the Demo Simulations section

5.1 Open-loop Dynamic Analysis

The principal advantage of creating simulation models of inverted pendula on a cart in form of an atomic icon is that detailed observation of their dynamics can be done with no other modeling than input/output block affiliation, for the model is complete and functional itself. In such a way, the analyses of the open-loop dynamical behavior of both the single and double inverted pendulum system were included in the *IPMaC* in form of demo schemes (*Single* and *Double Inverted Pendulum Response*), which were composed nearly exclusively of the *IPMaC* blocks. In both cases, the analysis of open-loop dynamic behavior was performed as a response to the *Impulse* block; to view the signals generated during simulation, *Scope* and *Scope rad2deg* blocks were used.

The default numeric values of system parameters are specified below. The dynamics of single inverted pendulum system was analyzed for two groups of parameters at once:

group I: $m_0 = 0.3\text{kg}$, $m_1 = 0.275\text{kg}$, $l_1 = 0.5\text{m}$, $\delta_0 = 0.3\text{kg s}^{-1}$, $\delta_1 = 0.01148\text{kg m}^2 \text{s}^{-1}$

group II: $m_0 = 0.1\text{kg}$, $m_1 = 1\text{kg}$, $l_1 = 0.8\text{m}$, $\delta_0 = 0.3\text{kg s}^{-1}$, $\delta_1 = 0.1\text{kg m}^2 \text{s}^{-1}$

and the default numeric values in the double inverted pendulum demo simulation are:

$m_0 = 0.3\text{kg}$, $m_1 = 0.275\text{kg}$, $m_2 = 0.275\text{kg}$, $l_1 = 0.5\text{m}$, $l_2 = 0.5\text{m}$, $\delta_0 = 0.3\text{kg s}^{-1}$, $\delta_1 = 0.1\text{kg m}^2 \text{s}^{-1}$, $\delta_2 = 0.1\text{kg m}^2 \text{s}^{-1}$.

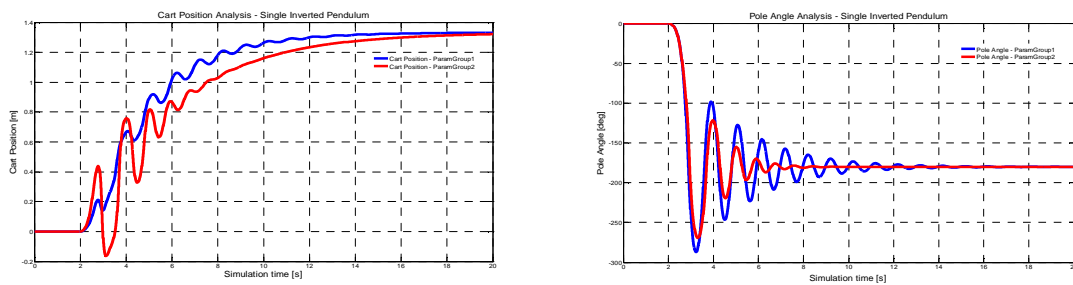


Fig. 8 Single Inverted Pendulum on a Cart - cart position and pendulum angle

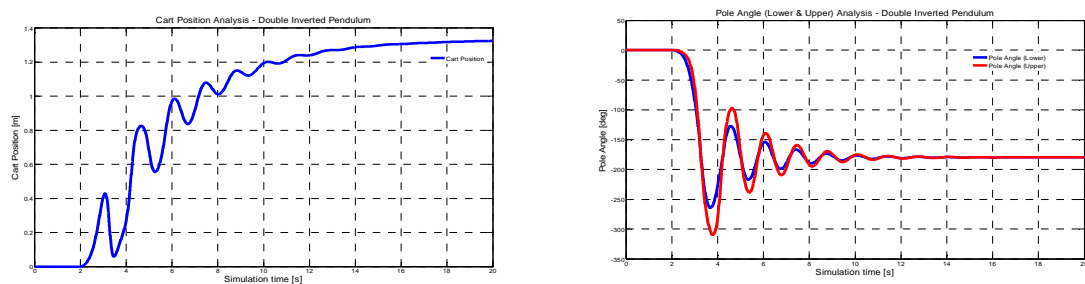


Fig. 9 Double Inverted Pendulum on a Cart – cart position, upper and lower pendula angles

It can be seen from the simulation results (Fig. 8, Fig. 9) that the generally known empirical observations of pendula behaviour are confirmed. Each pendulum of the system passes through oscillatory transient state until the system reaches the stable equilibrium point in which all pendula are pointing downward. The backward impact of the pendulum/pendula on the cart, which increases with the weight of the load, is also visible. We can therefore conclude that both models display acceptable overall performance.

5.2 Verification of state-space control algorithms

The example schemes below were picked from the *Demo Simulations* section to illustrate the way of interconnecting the blocks so as to control the single inverted pendulum system. A simulation scheme of a linearized and a nonlinear model is shown, with both the *State Space Controller* and *State Estimator* blocks as part of each scheme. The adjustment of the number of block inputs with respect to the tackled problem is also demonstrated.

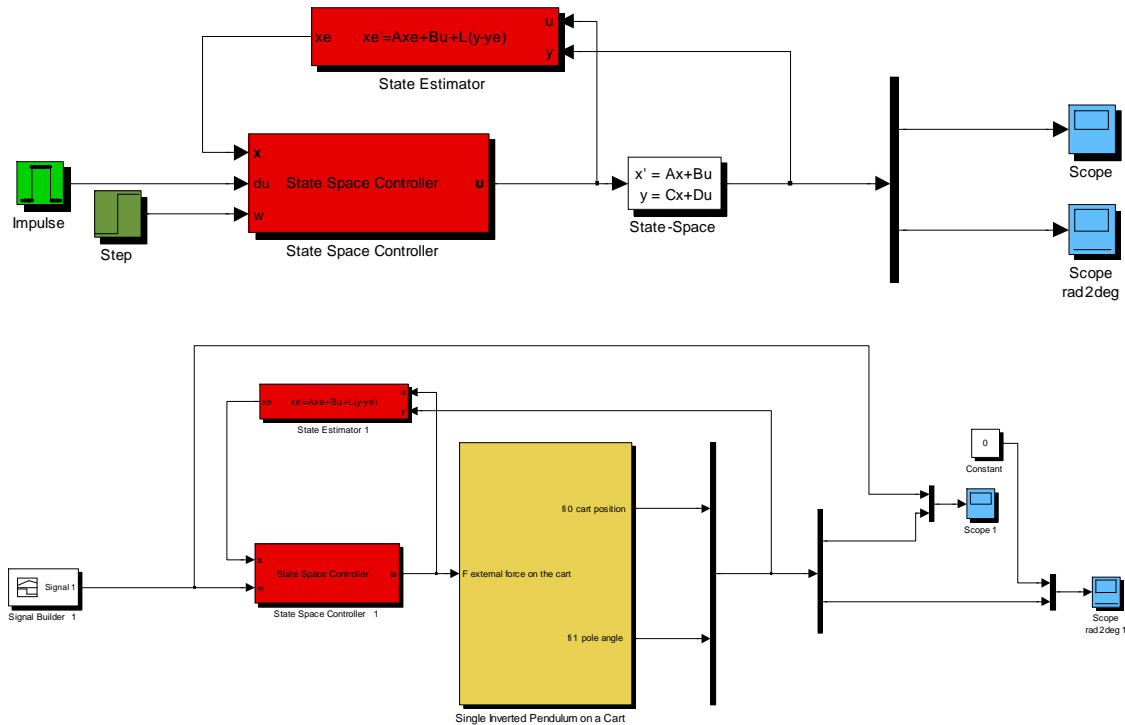


Fig. 10 Examples of control simulation schemes

Fig. 11 and Fig. 12 document the time behavior for both the cart position and the pendulum angle of the nonlinear single inverted pendulum system in case the control objective is to maintain the desired cart position while keeping the pendulum upright; no disturbance input is considered here. In order to use the linear methods of synthesis, the linearization of the nonlinear inverted pendulum system was required; this time the *matrices_single.m* was called with parameters in an *InitFcn* script.

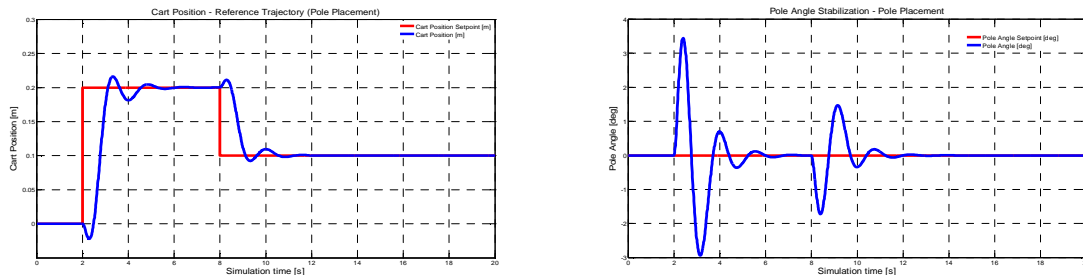


Fig. 11 Single Inverted Pendulum on a Cart – simulation results for pole-placement control without estimator (cart position, pendulum angle)

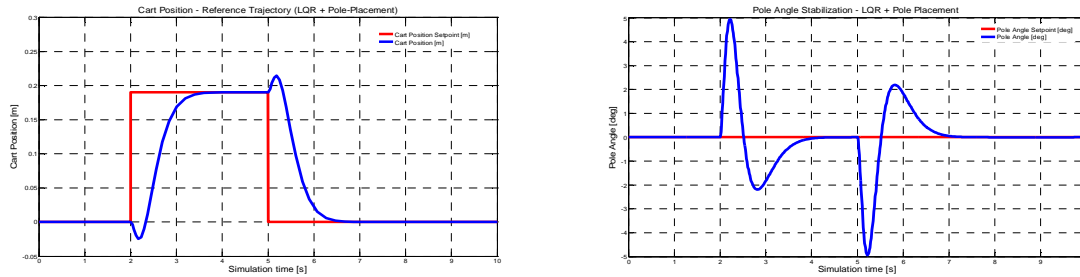


Fig. 12 Single Inverted Pendulum on a Cart – simulation results for LQR control with pole-placement-designed estimator (cart position tracking, pendulum angle stabilization)

The simulation results reveal that both control blocks do reasonably well. The ability of the designed blocks (*SSC* and *SE*) to control the system with respect to the all requirements presented above has been demonstrated for both methods of feedback gain design, although LQR control produces slightly better results despite the need for an estimator. Overall, the simulation results justify the use of linear control methods to control nonlinear systems.

6 Conclusion

The purpose of this paper was to propose an original conception of solving the task of modeling and control of inverted pendula systems. It focused on the description of the *Inverted Pendula Modeling and Control*, a custom-designed, structured *Simulink* block library, the core of which is represented by simulation models of a single and double inverted pendulum on a cart. The fact that the models have the form of dynamic-masked atomic library blocks (icons) allows for detailed observation of the systems' dynamics with no other modeling than input/output block affiliation. A set of suitable state-space control algorithms that stabilize the pendulum in the inverted (upright) position was designed and supported by a set of library blocks. Examples of open-loop dynamical analysis as well state-space controller design and verification can be run from the *Demo Simulations* section.

Practical importance of symbolic mathematical software was pointed out as *Symbolic Math Toolbox* was used in the process of development of a general procedure that returns the motion equations for a system of inverted pendula systems. Such automatic approach yields a particularly precise approximation of the real system's dynamics and eliminates any factual or numeric errors that should arise during mathematical modeling. The process of symbolic linear transformation of a specified nonlinear system in a user-chosen equilibrium point was equally algorithmized.

First published in June 2009 as an open system, the *IPMaC* is going through constant improvement process. Additions to sublibraries, as well as entire sections (e.g. a section on rotary pendula systems, where the base is moving in a plane rather than a single coordinate) are being designed. Above all, the control section is being expanded so as to include a wider variety of controller blocks and control schemes (e.g. cascade structures, feed-forward control, model predictive control, root-locus and frequency techniques) in addition to the already presented state-space feedback control algorithms. To enable further verification of the controllability properties of inverted pendula systems, the double inverted pendulum system is also planned to be included as the control plant. *IPMaC 2.0* is scheduled for 2010.

In summary, we believe that the idea of creating a thematic *Simulink* library, which would group accurate (based on extensive mathematical analysis) simulation models of mechanical systems together with useful input/output blocks; suitable controller blocks and demonstration simulations could find its use for a number of types of dynamical systems. Libraries of hydraulic, pneumatic or electrical systems could follow the steps of the *IPMaC*, which we consider to be a contribution to modeling and control education at technical universities.

Acknowledgements

This contribution is the result of the project implementation: *Centre of Information and Communication Technologies for Knowledge Systems* (project number: 26220120020), supported by the *Research and Development Operational Program* funded by the ERDF, and of the Vega project implementation: *Multiagent Network Control Systems with Automatic Reconfiguration* (No.1/0617/08), supported by the *Scientific Grant Agency of Slovak Republic*.

References

- [1] S. Jadlovská. *Inverted Pendula Modeling and Control [in Slovak]*. Bachelor thesis. Supervisor: doc. Ing. Anna Jadlovská, PhD., Technical University of Košice, Faculty of Electrotechnics a Informatics, Košice, 2009.
- [2] K. Sultan. Inverted Pendulum, Analysis, Design and Implementation, from <http://www.mathworks.com/matlabcentral/fileexchange/3790>, accessed on 1-8-2009
- [3] J. Roubal. *Nonlinear Pendulum Control [in Czech]*. Diploma thesis. Czech Technical University in Prague, Faculty of Electrical Engineering, Prague, 2002
- [4] M. Schlegel, J. Mešťánek. *Limitations on the Inverted Pendula Stabilizability According to Sensor Placement*. Proceedings of the 16th International Conference on Process Control, p. 54, 2007
- [5] A. Bogdanov. *Optimal Control of a Double Inverted Pendulum on the Cart*. Technical Report CSE-04-006, OGI School of Science and Engineering, OHSU, 2004
- [6] M. Demirci. *Design of Feedback Controllers for a Linear System with Applications to Control of a Double-Inverted Pendulum*. International Journal of Computational Cognition, Vol. 2, No. 1, p. 65-84, 2004
- [7] A. Jadlovská. *Modeling and Control of Dynamic Processes Using Neural Networks*. Informatech, Košice,

Bc. Slávka Jadlovská
Department of Cybernetics and Artificial Intelligence
Faculty of Electrical Engineering and Informatics
Technical University of Košice
Letná 9, 04200 Košice, Slovak Republic
e-mail: sjadlovska@gmail.com

doc. Ing. Anna Jadlovská, PhD.
Department of Cybernetics and Artificial Intelligence
Faculty of Electrical Engineering and Informatics
Technical University of Košice
Letná 9, 04200 Košice, Slovak Republic
e-mail: Anna.Jadlovska@tuke.sk